Immersive Image-Based Modeling of Polyhedral Scenes

Gilles Simon

Immersive Image-Based Modeling of Polyhedral Scenes

Gilles Simon*
Nancy University - INRIA Project-Team MAGRIT

ABSTRACT

In this paper, we describe a purely image-based system that allows a user to interactively capture the 3D geometry of a polyhedral scene with the aid of its physical presence. A video camera is used as both an interaction and tracking device. The 3D user interface is intuitive to a non-expert and the mouseless control procedure makes the system particularly suitable for mobile devices such as PDAs and mobile phones. The efficiency and accuracy of the method are demonstrated on a polyhedral scene made of two house-like boxes.

Keywords: Augmented Reality, Image-Based Modeling, Construction at a Distance, 3D User Interfaces, Wearable Computers

Index Terms: I.2.10 [Artificial Intelligence]: Vision and Scene Understanding—Modeling and recovery of physical attributes I.3.6 [Computer Graphics]: Methodology and Techniques—Interaction techniques

1 INTRODUCTION

Acquiring the 3D geometry of arbitrary scenes has been a primary objective of both the computer vision and graphics communities for many decades. Applications are numerous in various domains such as construction, GIS and 3D maps, virtual tours, visual effects and AR. Existing modeling methods usually rely on two separate stages: first, some data about the scene (photographs, videos, laser measurements, ...) are acquired on-site. Then these data are treated off-line using some specific manipulations and algorithms. Unfortunately, this process can be time-consuming and tedious; moreover, there is no guarantee after the first stage that the required model is fully extractable from the acquired data and additional acquisitions are sometimes needed to supplement the missing parts.

This paper tries to bridge the gap between data acquisition and their exploitation. A pioneer work has been described in [2], where mobile AR users are enabled to interactively capture planar shaped objects with the aid of their physical presence. Users interact with the computer using a set of pinch gloves and the camera is tracked using an inertial sensor and a GPS. Our system differs from that work in that the only device required for both tracking and interacting is a handheld camera coupled with three keyboard keys (e.g. a mobile phone); this makes it suitable for indoor as well as outdoor use. A purely image-based interactive model building system has recently been presented in [1]. This work has several common aspects with our work, such as the use of a camera-mouse and model-based tracking. However, the modeling task is less constrained in our system in that it does not require any initial template to be placed in the scene, nor that the current model partly stays in the camera field of view while new structures are added. Moreover, defining a vertex in [1] is based on parallax motion which is not well appropriate for modeling objects far from the camera.

Finally, our system can be seen as an immersive version of the widely used 3D drawing software SketchUp™ (http://sketchup.google.com). This tool combines some of the features of pencil-and-paper sketching and some of the features of CAD systems to provide a lightweight, gesture-based interface for 3D polyhedral modeling. In the latest releases of SketchUp, the user is able to align the world axes to match a photo perspective. With this done, he can create models using the photo as a direct reference; mouse strokes are converted into 3D-space using inverse ray intersections with the previously defined geometry (or the ground plane by default). In addition, a virtual camera can be manipulated in order to move around the 3D model under construction. All these principles have been taken up in our implementation, but with the important difference that we consider dynamic video images instead of static ones; moreover, the video camera is used as the interaction device instead of a mouse and virtual camera manipulations are replaced by real ones.

*e-mail: Gilles.Simon@loria.fr

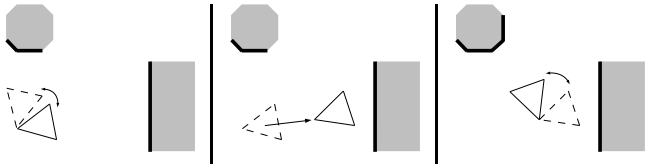

Figure 1: The system alternates between modeling operations (only pure rotations of the camera are allowed) and 6-DOF camera tracking. Thick lines represent the modeled faces of the scene.

2 OVERVIEW OF THE SYSTEM

Scene modeling is done by alternating modeling interaction and camera moving sequences (Fig. 1):

Modeling mode. Pure rotations are applied to the camera (e.g. when using a head-mounted display, this amounts to look around the scene by only turning the head). Using the camera as an interaction device, the user is able both to calibrate the camera and to define the scene geometry. It must be noticed that this mode may be used alone, providing already interesting contributions to classical single view metrology, as (i) by turning the head the reachable field of view is much larger than the camera's field of view and (ii) mouse manipulations are replaced by camera manipulations, which can be of great interest in a mobile context.

Tracking mode. When at least one face of the scene has been described and the camera undergoes a general motion, a 6-DOF (degrees of freedom) camera tracking is performed, based on the available geometry [3]. This allows the user to get closer to some parts of the scene or make some new faces visible before continuing modeling. In this mode, the geometry previously modeled has to stay (at least partly) visible in the camera field of view, which is not required in modeling mode. A pose recovery procedure based on SIFT features can be called upon user request or each time a tracking failure is detected by the system.

The system starts in modeling mode. Once a 3D geometry has been initialized, switches between modeling and tracking modes are done automatically by the system, depending on the motion applied to the camera (Akaike's motion model selection is used [3]).

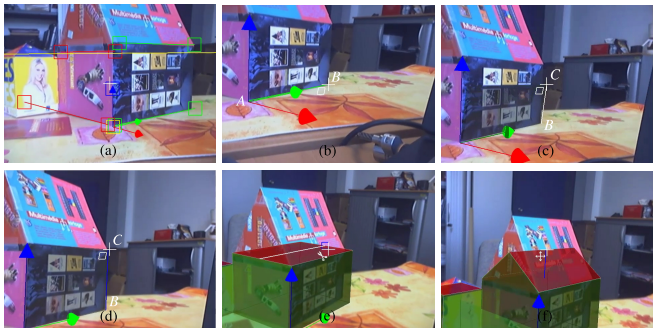

Figure 2: Snapshots of the system in use.

3 Modeling Interactions

A key idea in this work is that the user can interact with the images using the video camera itself. Indeed, let us consider that he/she wants to draw a line stroke on a video image between two physical points A and B of a real-world scene; there are two possible ways to do this: (i) the camera stays fixed and a mouse is used to move a cursor to the two respective endpoints; (ii) the camera is rotated so that the physical points apparently move to a fixed cursor, for instance at the center of the image. The second solution, which is used in our implementation, does not require mouse support. However, the position of point A has to be updated from frame two frame while point B is aimed at. Fortunately, camera rotations only induce homographic deformations of the image, that can be computed easily e.g. using keypoint matches. In both cases a click has to be done when the cursor (or the camera) is correctly positioned. In the second solution, mouse clicks can be replaced by key presses or any other input controls such as buttons, voice commands, etc. In our system, only three keys are used for all operations: one to "click" or "drag and drop" in the image, a second to cancel the current operation or request a pose recovery and a third to scroll the tools menu.

Before being able to model the scene, the user has to calibrate the camera: this is done by indicating two sets of horizontal parallel lines orthogonal to each other (Fig. 2, frame a). In addition, the user can drag and drop the origin of the world frame, providing a ray of possible 3D positions of that point. The depth of the origin is chosen so that the unit up vector has an arbitrary size in the image plane. The user can also change the scale of the scene by moving the extremity of the unit up vector. Once the camera parameters are known, the user can start modeling. New faces are instantiated in contact with existing faces or the ground plane by default. Contacts are guaranteed by "attracting" the clicked points to existing 3D points which appear in a specific color when close to the cursor: yellow for a face vertex, blue for the middle of an edge and red for the closest point on an edge. Faces under the cursor are also specifically outlined and new 3D vertices can be generated by using inverse ray intersection. In addition, most actions are guided using a line stroke, which itself can be attracted by the projected world axes.

Six different tools have been implemented which are summarized in Tab. 1. Fig. 2 illustrates how a simple house-like box can be modeled using the Add, Extrude, Cut and Move tools: a rectangular face is first created using two line strokes AB and BC: according to the priority rules described at bottom of Tab. 1, AB is instantiated in the ground plane and parallel to the green axis (frame b); then C is dragged in the plane orthogonal to AB (frame c) and finally such that BC is aligned with the blue axis (frame d); the new face is then extruded forming a box (frame e) whose top face is subdivided so that the cut line joins the middles of two opposite edges; this line is moved along the blue axis in order to form the roof (frame f).

□	Add a rectangular face by clicking three points $A,B,C^{(*)}$. The rectangle is generated in the plane ABC using A and C as diagonally opposite vertices.
	Extract texture and tracking features from the selected face.
	Extrude the selected face.
	Move the selected vertex, edge or face.
	Subdivide the selected face by joining two of its vertices.
×	Delete the selected face.

$^{(*)}$ The inverse ray intersection priorities are:
A: selected vertex > selected face > ground
B: selected vertex > projected world axis $\parallel AB$ > plane(A)
C: selected vertex > projected world axis $\parallel BC$ > plane $\perp [AB]$

Table 1: Manipulation tools used in our prototype.

4 Evaluation

A video is associated with this paper showing the system in action (http://www.loria.fr/~gsimon/ismar09/). A Dell Precision M6300 laptop coupled with a simple Logitech webcam were used for this experiment. The system runs at video rate in standard mode and 2 to 6Hz when the recovery procedure is called. A simple polyhedral scene, made of two house-like boxes (Fig. 2), is modeled. Several modeling and tracking sequences alternate during a 6-minute working session (the video has been cut and accelerated 1.5×). Note that it can be easily distinguished when the system is running in modeling mode and when it is running in tracking mode, as the cross cursor disappears in the second case.

Fig. 3 shows the errors obtained on the recovered 3D geometry. The largest relative error (12.3%, 9 mm) actually corresponds to a depth inaccuracy on d_6; getting closer to the related physical face before modeling it or making it appear more "fronto-parallel" would probably have led to better accuracy.

	Rec.	Exp.	% Error
d_1	11.5	11.5	Ref.
d_2	19.2	19.5	-1.3
d_3	14.0	14.5	-3.3
d_4	14.1	15.0	-6.0
d_5	7.9	7.7	2.8
d_6	6.5	7.4	-12.3
d_7	9.5	9.7	-2.3
d_8	5.4	5.9	-8.1

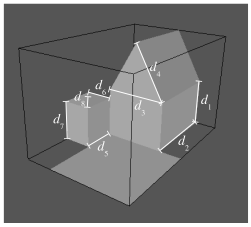

Figure 3: Recovered/expected distances (in cm) and percentage errors obtained on the recovered geometry.

5 Conclusion

We regard this work as a proof-of-concept application. Its main flaw resides in that 6-DOF tracking is only possible when a face of the modeled scene is visible; this can make it difficult to go "behind" the modeled scene. To tackle this issue, two different approaches may be considered: the first one may consist in allowing the user to model different parts of the scene independently each other and merge these parts later using a specific tool; the second approach may be to automatically reconstruct 3D points out of the model faces during 6-DOF tracking and use these points to track the camera when no face is visible or in addition to the visible faces; this may be done within a SLAM framework.

References

[1] P. Bunnun and W. W. Mayol-Cuevas. OutlinAR: an assisted interactive model building system with reduced computational effort. In ISMAR, pages 61–64, Los Alamitos, CA, USA, 2008.

[2] W. Piekarski and B. H. Thomas. Tinmith-Metro: New Outdoor Techniques for Creating City Models with an Augmented Reality Wearable Computer. In ISWC, pages 31–38, Zurich, Switzerland, 2001.

[3] F. Vigueras, M.-O. Berger, and G. Simon. Iterative multi-planar camera calibration: Improving stability using model selection. In VVG, Bath, United Kingdom, 2003.